**# Purchase Management System**

This system allows users to make purchases and retrieve their purchase history. It consists of a customer-facing API, a customer management API, and utilizes Kafka for message queuing and MongoDB for data storage.

**## Installation and Setup**

**### 1. Customer Facing API**

Install the customer-facing API:

**helm install customer-facing-api ./customer-facing-api**

This API handles user requests and communicates with Kafka and the customer management API.

**Key functionalities:**

Exposes a POST /buy endpoint to process purchase requests

Sends purchase data to Kafka

Exposes a GET /getAllUserBuys endpoint to retrieve all user purchases

Communicates with the customer management API to fetch purchase data

2. Customer Management API

Install the customer management API:

**helm install customer-management-api ./customer-management-api**

This API handles data management and interacts with Kafka and MongoDB.

Key functionalities:

Consumes purchase messages from Kafka

Stores purchase data in MongoDB

Exposes a GET /purchases endpoint to retrieve all purchases

**3. MongoDB**

Install MongoDB:

**helm repo add bitnami https://charts.bitnami.com/bitnami**

**helm install my-mongodb bitnami/mongodb --set auth.enabled=false**

MongoDB is used to store the purchase data.

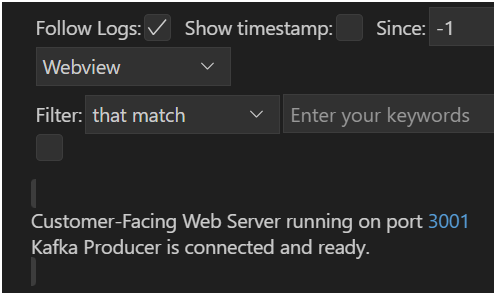
**4. Kafka**

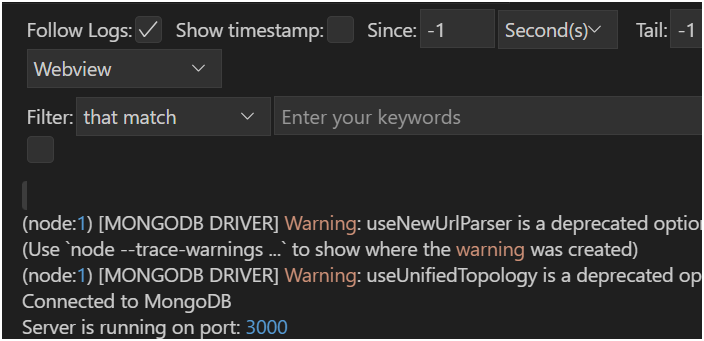
Install Kafka:

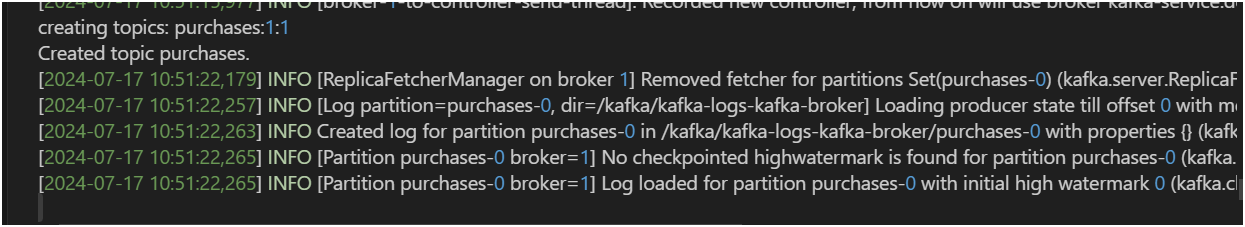
**helm install my-kafka ./kafka-helm**

Kafka is used for message queuing between the customer-facing API and the customer management API.

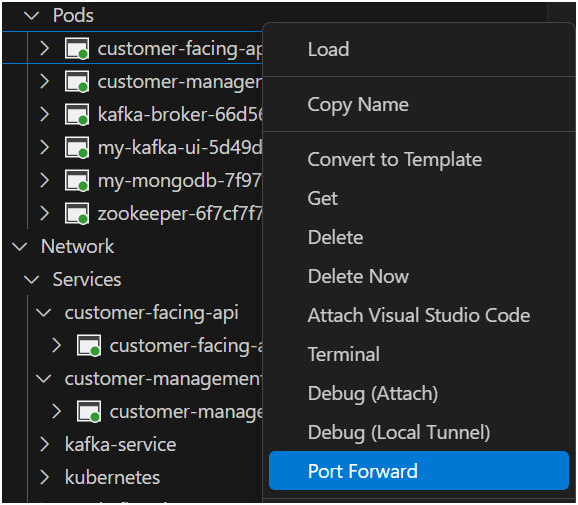
**Verification and Testing**

Check the logs of all components to ensure there are no errors and all connections are established.  
  






Port forward the customer-facing API:



kubectl port-forward service/customer-facing-api 3001:3001

**Test the purchase functionality:**

Use Postman to send a POST request to http://localhost:3001/buy

Include JSON body: {"username":"shuki","userid":"1","price":"100","timestamp":"10:00"}

**Behind the scenes:**

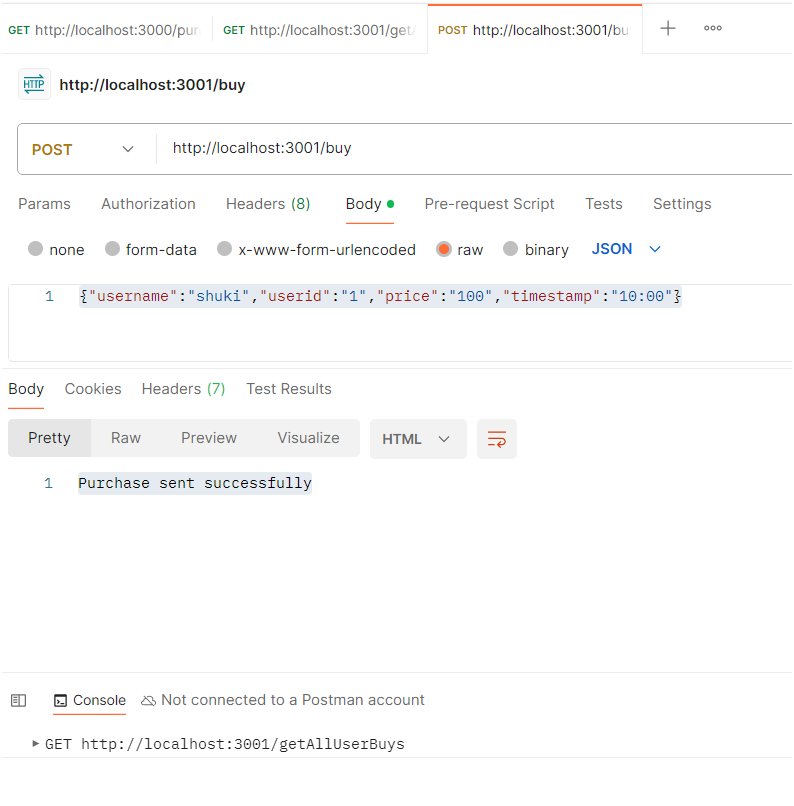
The customer-facing API receives the request

It sends the purchase data to Kafka

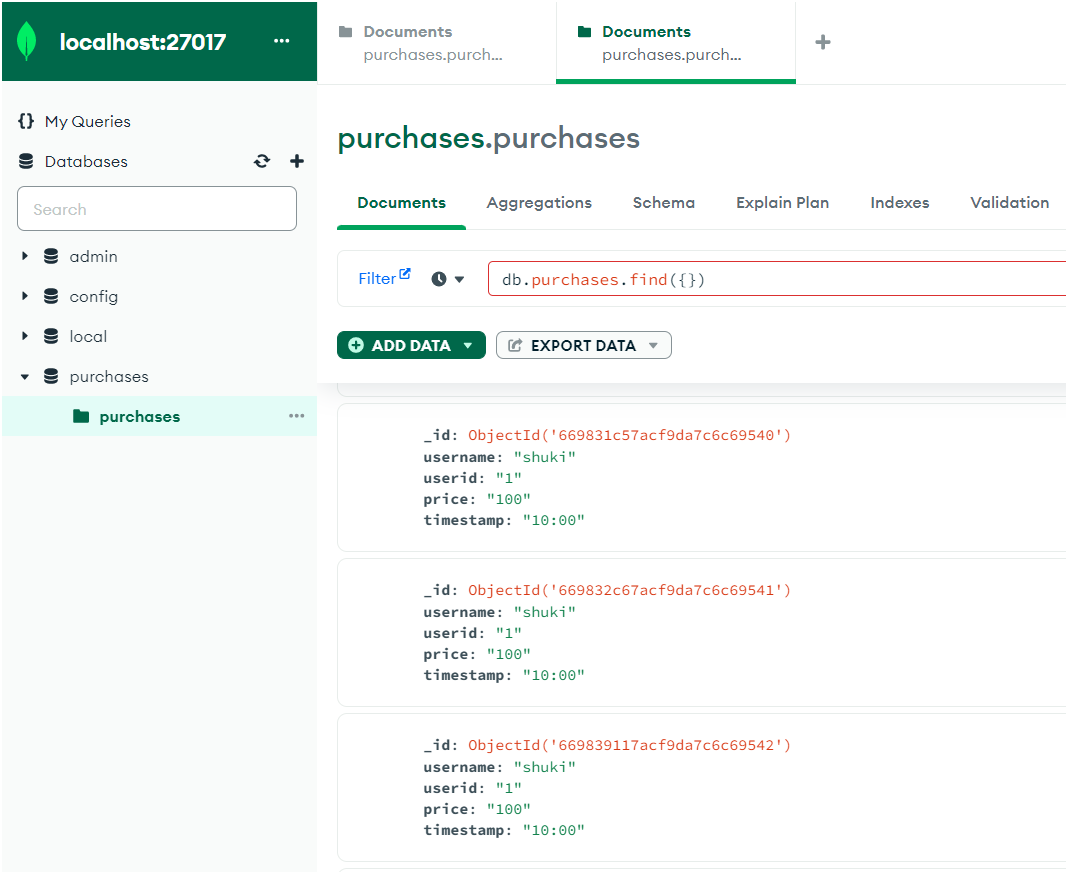
The customer management API consumes the message from Kafka

The purchase data is then stored in MongoDB

You should receive a 200 status code with the message "Purchase sent successfully".



**Verify data in MongoDB:**

Port forward MongoDB and check that the purchase data has been created  
  


Test retrieving all user purchases:

Send a GET request to http://localhost:3001/getAllUserBuys

Behind the scenes:

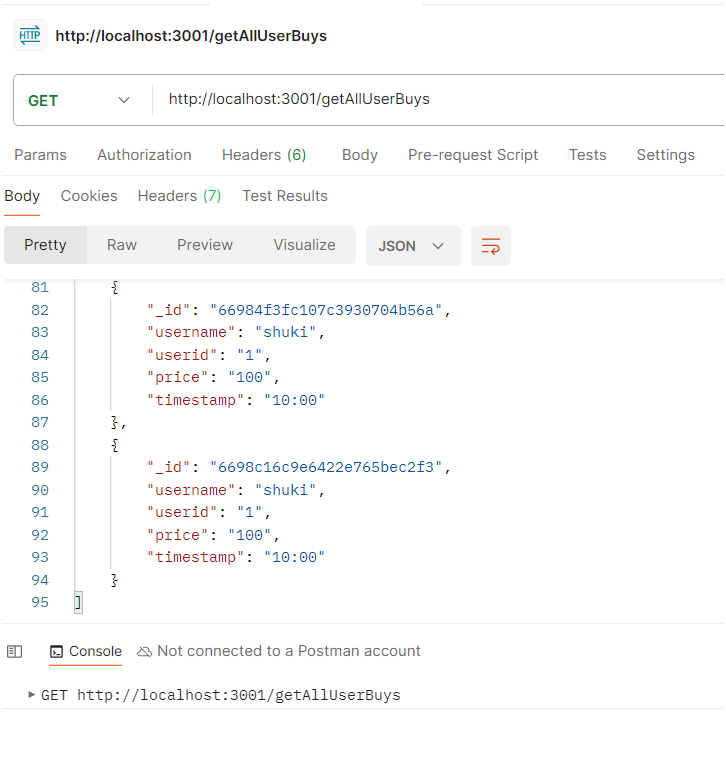
The customer-facing API receives the request

It forwards the request to the customer management API

The customer management API retrieves all purchases from MongoDB

The data is sent back through the customer-facing API to the client

You should receive a JSON array containing all user purchases.



Troubleshooting

If you encounter any issues:

Check the logs of each component for error messages

Ensure all services are running and properly connected

Verify that Kafka and MongoDB are accessible from both APIs